Una debilidad de la estrategia Uno-contra-Uno en clasificación: Potenciando las clases difíciles

Mikel Galar[1], Alberto Fernández[2], Edurne Barrenechea[1], and Francisco Herrera[3]

[1] Departamento de Automática y Computación,
Universidad Pública de Navarra, 31006, Pamplona, España
{mikel.galar,edurne.barrenechea}@unavarra.es
[2] Departamento de Ciencias de la Computación,
Universidad de Jaén, 23071, Jaén, España
alberto.fernandez@ujaen.es
[3] Departamento de Ciencias de la Computación e Inteligencia Artificial,
Universidad de Granada, 18071, Granada, España
herrera@decsai.ugr.es

Resumen La estrategia de binarización Uno-contra-Uno es uno de los métodos más utilizados a la hora de afrontar problemas con múltiples clases. En este trabajo analizamos el hecho de que en esta estrategia se favorece a las clases fáciles en perjuicio de las difíciles (aquellos conceptos del problema que son más difíciles de identificar y que por tanto, obtienen un menor ratio de acierto). Además, presentamos una posible solución para potenciar las clases difíciles mediante una generalización del voto ponderado. Esta propuesta permite obtener una clasificación más balanceada sobre todas las clases, sin acarrear una pérdida de la precisión global.

Palabras clave: multi-clase, uno-contra-uno, descomposición, clases difíciles

1. Introduction

Las estrategias de descomposición [13] permiten afrontar problemas multi-clase mediante clasificadores binarios. Entre ellas, una de las estrategias más utilizadas es la estrategia Uno-contra-Uno (*One-vs-One*, OVO), donde se divide el problema multi-clase original en tantos nuevos subproblemas como posibles pares de clases podamos obtener. Cada nuevo subproblema binario es afrontado por un clasificador independiente, cuya salida es posteriormente combinada con la del resto para obtener la clase final con la que etiquetar una nueva instancia [9].

En los problemas de clasificación, las características que definen a cada una de las clases suelen ser, en general, diferentes: distribución de ejemplos, relación entre las clases, relación entre los ejemplos de la propia clase o solapamiento entre clases. Como consecuencia, algunas de las clases pueden ser más difíciles de distinguir que otras. En este contexto, podemos definir las *clases difíciles*

como aquellas sobre las que se obtiene un ratio de acierto menor (más bajo que el obtenido por el resto de las clases).

En esta contribución nos centramos en aquellos problemas en los cuales todas las clases son igualmente importantes. La precisión global no es capaz de valorar igualmente a todas las clases, ya que los errores sobre las clases difíciles se ven contrarrestados por los aciertos sobre las clases más sencillas, incluso aunque el conjunto de datos esté balanceado. Esto es debido a que esta medida realiza una media sobre todas las instancias, sin tener en cuenta la precisión sobre cada una de las clases independientemente. Por ello, en este trabajo vamos a considerar otras medidas que nos permitan evaluar correctamente la precisión sobre todas las clases de manera global.

Nuestro objetivo en este trabajo es doble: 1) tratar de explicar el porqué de la debilidad de la estrategia OVO cuando tratamos de obtener una buena predicción sobre todas las clases; 2) introducir un nuevo modelo de agregación basado en Funciones de Equivalencia Restringida (*Restricted Equivalence Functions*, REFs) [5], que nos permita modificar modificar las fronteras de decisión de los clasificadores base y potenciar la clasificación de las clases difíciles, sin necesidad de cambiar los clasificadores base entrenados.

Los experimentos que llevamos a cabo en este trabajo incluyen veintiocho conjuntos de datos de los repositorios UCI [3] y KEEL [1]. Utilizamos, además de la precisión global, otras medidas que tienen en cuenta a las clases difíciles y contrastamos los resultados obtenidos mediante test estadísticos [6,10]. Debido a las limitaciones de espacio, utilizaremos únicamente como clasificador base las máquinas vector soporte (*Support Vector Machine*, SVM) [15].

El resto del trabajo está organizado como sigue. En la Sección 2 analizamos el problema de las clases difíciles en la estrategia OVO. Posteriormente, en la Sección 3 mostramos nuestra propuesta para potenciar las clases difíciles. El marco experimental utilizado está explicado en la Sección 4. En la Sección 5 presentamos el estudio experimental que hemos llevado a cabo y finalmente en la Sección 6 presentamos las conclusiones de este trabajo.

2. El problema de las clases difíciles en OVO

Esta sección recuerda el funcionamiento de la estrategia OVO (Subsección 2.1), y explicamos el problema de las clases difíciles (Subsección 2.2).

2.1. Descomposición Uno-contra-Uno

OVO divide un problema de m clases en $m(m-1)/2$ subproblemas binarios (todos los posibles pares de clases) que son afrontados por clasificadores base independientes. Una nueva instancia se clasifica considerando la salida de cada uno de los clasificadores. Cada clasificador que distingue un par de clases $\{C_i, C_j\}$ devuelve un grado de confianza $r_{ij} \in [0, 1]$ en favor de la clase C_i ($r_{ji} = 1 - r_{ij}$). Estas salidas pueden almacenarse en lo que se denomina matriz de votos de la

siguiente forma:

$$R = \begin{pmatrix} - & r_{12} & \cdots & r_{1m} \\ r_{21} & - & \cdots & r_{2m} \\ \vdots & & & \vdots \\ r_{m1} & r_{m2} & \cdots & - \end{pmatrix} \tag{1}$$

Para establecer la clase a la que pertenece la instancia puede usarse cualquiera de las agregaciones existentes en la literatura [9] sobre la matriz de votos. La más simple es la estrategia del voto, donde cada clasificador da un voto para la clase que predice y la clase con más votos es con la que se etiqueta la instancia.

2.2. Una debilidad de la estrategia Uno-contra-Uno

La mejora en términos de precisión obtenida por la estrategia OVO frente a los clasificadores base [9] se debe en gran medida a la mejora en la precisión obtenida sobre las clases más fáciles. A continuación, tratamos de mostrar la razón por la que la estrategia OVO tiende a mejorar la precisión sobre las clases fáciles, sin mejorarla sobre las difíciles. Para ello, consideramos el escenario más simple: OVO con la estrategia del voto simple. Recordamos que el ratio de verdaderos positivos (*True Positive Rate*, TPR) de una clase es el número de ejemplos correctamente clasificados de dicha clase entre el total de ejemplos de la misma.

Enunciado del problema y notación:

- Sea un problema con m clases, $\mathbb{C} = \{C_1, \ldots, C_m\}$. Una de ellas ($C_d$) es mucho más difícil de clasificar. El resto de clases $\mathbb{C}_e = \{C_{e_1}, C_{e_2}, \ldots, C_{e_{m-1}}\}$ son igualmente distinguibles.
- Sea TPR_d el TPR sobre la clase difícil y $\mathrm{TPR}_e = \mathrm{TPR}_{e_1} = \ldots = \mathrm{TPR}_{e_{m-1}}$ el TPR sobre cada una de las clases fáciles. Por tanto, $\mathrm{TPR}_d < \mathrm{TPR}_e$.

Suposiciones:

1. Todos los TPR son iguales en todos los clasificadores base que consideran la misma clase (suposiciones similares son consideradas en [12]).
2. Independencia de los clasificadores base (se supone en OVO).
3. Una instancia se clasifica correctamente si todos los clasificadores base competentes (aquellos que consideran la clase real en el entrenamiento) clasifican correctamente la instancia.

En este marco de trabajo, una instancia $\mathbf{x}_{e_1}$ de una clase fáciles (C_{e_1}) va a ser clasificada. La probabilidad de clasificarla correctamente $P(h_{ovo}(\mathbf{x}_{e_1}) = C_{e_1})$ (donde h_{ovo} es el clasificador OVO) viene dada por el TPR de cada uno de los $m-1$ clasificadores competentes para dicha instancia. Por tanto $P(h_{ovo}(\mathbf{x}_{e_1}) = C_{e_1}) = \prod_{i=1}^{m-1} \mathrm{TPR}_{e_1}$. De manera similar, la probabilidad de clasificar correctamente una instancia $\mathbf{x}_d$ de la clase difícil (C_d) es $P(h_{ovo}(\mathbf{x}_d) = C_d) = \prod_{i=1}^{m-1} \mathrm{TPR}_d$.

Problema: Dado que estas probabilidades se calculan como el producto de los TPRs individuales de los clasificadores competentes, según aumenta el número de clases m, la probabilidad de clasificar correctamente la instancia de

la clase difícil decrece más rápidamente, mientras que la probabilidad sobre la clasificación de instancias del resto de clases no decrece tan drásticamente. Como consecuencia, la probabilidad de clasificar correctamente una instancia de la clase difícil es proporcionalmente mucho menor:

$$P(h_{ovo}(\mathbf{x}_{e_1}) = C_{e_1}) = \prod_{i=1}^{m-1} \text{TPR}_{e_1} \gg \prod_{i=1}^{m-1} \text{TPR}_d = P(h_{ovo}(\mathbf{x}_d) = C_d),$$

donde $\gg$ indica que la parte derecha es proporcionalmente mucho menor según aumenta m, a pesar de que las dos partes decrecen. La Tabla 1 muestra claramente este hecho, donde observamos la evolución de la probabilidades de clasificar correctamente una instancia con un TPR específico para la clase en cada clasificador base. Lógicamente, estos porcentajes no son más que cotas inferiores de la probabilidad debido a las suposiciones impuestas.

Tabla 1. Probabilidad de clasificar correctamente una instancia de una clase con un TPR específico y diferentes números de clases.

Clases	TPR 0,5	TPR 0,55	TPR 0,6	TPR 0,65	TPR 0,7	TPR 0,75	TPR 0,8	TPR 0,85	TPR 0,9	TPR 0,95	TPR 1,0
2	0,5	0,55	0,6	0,65	0,7	0,75	0,8	0,85	0,9	0,95	1,0
3	0,250	0,303	0,360	0,423	0,490	0,563	0,640	0,723	0,810	0,903	1,0
4	0,125	0,166	0,216	0,275	0,343	0,422	0,512	0,614	0,729	0,857	1,0
5	0,063	0,092	0,130	0,179	0,240	0,316	0,410	0,522	0,656	0,815	1,0
6	0,031	0,050	0,078	0,116	0,168	0,237	0,328	0,444	0,591	0,774	1,0
7	0,016	0,028	0,047	0,075	0,118	0,178	0,262	0,377	0,531	0,735	1,0
8	0,008	0,015	0,028	0,049	0,082	0,134	0,210	0,321	0,478	0,698	1,0
9	0,004	0,008	0,017	0,032	0,058	0,100	0,168	0,273	0,431	0,663	1,0
10	0,002	0,005	0,010	0,021	0,040	0,075	0,134	0,232	0,387	0,630	1,0
11	0,001	0,003	0,006	0,014	0,028	0,056	0,107	0,197	0,349	0,599	1,0
12	0,0005	0,001	0,004	0,009	0,020	0,042	0,086	0,167	0,314	0,569	1,0

¿Cómo puede aliviarse este problema?

1. Mejorando el TPR_d en cada uno de los clasificadores base.
2. Usando una agregación que tenga en cuenta el problema de las clases difíciles sin necesidad de alterar los clasificadores base que estan debajo.

El primer caso es la solución directa al problema, pero hay que tener en cuenta que, incluso aumentando el TPR_d, mientras los TPRs sobre todas las clases sean diferentes, la dificultad se mantendrá. Además, dicha mejora no es fácil de obtener. Por estas razones, nos centramos en el segundo caso, que de cualquier modo, podría combinarse con el primero.

Desde nuestro punto de vista, las matrices de votos contienen información suficiente como para obtener resultados significativamente diferentes sobre las clases difíciles cambiando únicamente la agregación. Por esta razón, las matrices de votos utilizadas en los experimentos serán las mismas para todas las agregaciones. Por tanto, todas las diferencias mostradas en el apartado experimental serán debido a las agregaciones y no a los clasificadores base, lo cual es de gran importancia para poder evaluar debidamente el rendimiento de la metodología que proponemos.

3. Una agregación basada en REFs

En esta sección introducimos el nuevo modelo de agregación para OVO. Primero recordamos varios conceptos preliminares en la Subsección 3.1, que son

necesarios para poder presentar posteriormente el nuevo modelo en la Subsección 3.2.

3.1. Funciones de Equivalencia Restringidas

Para poder introducir el nuevo modelo de agregación necesitamos recordar varios conceptos provenientes de la teoría de lógica difusa. En la teoría difusa, una negación modela el concepto de complementario u opuesto:

Definición 1 *Una función* $n : [0,1] \to [0,1]$ *con* $n(0) = 1$, $n(1) = 0$, *estrictamente decreciente y continua es una negación estricta. Además, si* n *es involutiva, es decir, si* $n(n(x)) = x$ *para todo* $x \in [0,1]$, *entonces* n *es una negación fuerte.*

Las REFs [5] permiten medir lo próximos que son dos valores puntuales.

Definición 2 *[5] Una función* $\mathrm{REF} : [0,1]^2 \to [0,1]$ *se dice función de equivalencia restringida asociada a una negación fuerte* n *si satisface las siguientes condiciones*

1. $\mathrm{REF}(x,y) = \mathrm{REF}(y,x)$ *para todo* $x, y \in [0,1]$;
2. $\mathrm{REF}(x,y) = 1$ *si y solo si* $x = y$;
3. $\mathrm{REF}(x,y) = 0$ *si y solo si* $x = 1$ *e* $y = 0$ *o* $x = 0$ *e* $y = 1$;
4. $\mathrm{REF}(x,y) = \mathrm{REF}(n(x), n(y))$ *para todo* $x, y \in [0,1]$;
5. *Para todo* $x, y, z \in [0,1]$, *si* $x \leq y \leq z$, *entonces* $\mathrm{REF}(x,y) \geq \mathrm{REF}(x,z)$ *y* $\mathrm{REF}(y,z) \geq \mathrm{REF}(x,z)$.

En este trabajo, el interés de esta medida de proximidad reside en la posibilidad de parametrizarla por medio de automorfismos de la siguiente forma.

Definición 3 *Una función* $\varphi : [a,b] \to [a,b]$ *continua y estrictamente creciente tal que* $\varphi(a) = a$ *y* $\varphi(b) = b$ *se denomina automorfismo del intervalo* $[a,b] \subset \mathbb{R}$.

Proposición 1 *[5] Sean* φ_1, φ_2 *dos automorfismos del intervalo* $[0,1]$. *Entonces* $\mathrm{REF}(x,y) = \varphi_1^{-1}(1 - |\varphi_2(x) - \varphi_2(y)|)$ *es una función de equivalencia restringida asociada a la negación fuerte* $n(x) = \varphi_2^{-1}(1 - \varphi_2(x))$.

Una forma sencilla de parametrizar los automorfismos es mediante un parámetro: sea $\varphi(x) = x^\lambda$, y por tanto, $\varphi^{-1}(x) = x^{1/\lambda}$, con $\lambda \in (0, \infty)$.

3.2. Una generalización del método del voto ponderado

La agregación que proponemos en este trabajo es una generalización del método del voto ponderado (*Weighted Voting*, WV), cuya robustez ha sido tanto teórica como empíricamente probada [11]: $\mathrm{Clase} = \arg \max_{i=1,\ldots,m} \sum_{1 \leq j \neq i \leq m} r_{ij}$.

En nuestro modelo de agregación, en vez de sumar directamente las confianzas dadas por los clasificadores, primero comparamos estas confianzas con el voto seguro (es decir, 1), ya que es el caso en el que debemos dar el voto

más alto. Por tanto, cuanto más similar sea r_{ij} al 1, más importancia tendrá el voto, tal y como ocurría hasta ahora. Sin embargo, nuestra intención es que la importancia del voto dado por cada clasificador sea diferente. Por ello, comparamos ambos valores mediante una REF, $\text{REF}(r_{ij}, 1)$, que nos proporciona una medida de cómo de cerca está r_{ij} del voto seguro. Partiendo de la Proposición 1 y considerando $\varphi_1 = x^{\lambda_1}$, $\varphi_2 = x^{\lambda_2}$, las operaciones y parámetros necesarios para llevar a cabo la comparación pueden reducirse:

$$\text{REF}(x, 1) = (1 - |x^{\lambda_2} - 1^{\lambda_2}|)^{1/\lambda_1} = (x)^{\lambda_2/\lambda_1} = x^{\lambda} \tag{2}$$

La forma en la que podemos alterar la importancia de los votos puede verse en la Figura 1. Esta figura muestra la influencia de λ en la aplicación de la REF en la comparativa frente al voto seguro. Es interesante observar que $\lambda = 1$ no modifica el voto del clasificador, ya que $\text{REF}(r_{ij}, 1) = r_{ij}$, mientras que valores de $\lambda < 1$ potencian el voto ($\text{REF}(r_{ij}, 1) > r_{ij}$) y ocurre lo contrario con $\lambda > 1$ ($\text{REF}(r_{ij}, 1) < r_{ij}$).

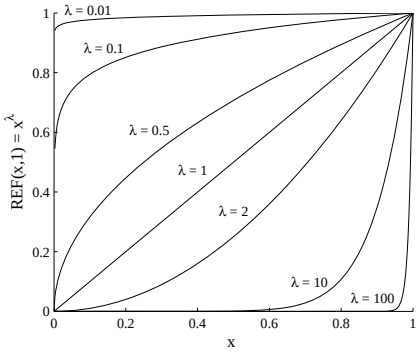

Figura 1. Influence of the parameter λ in $\text{REF}(x, 1)$.

Nota *Por tanto, atendiendo a la Figura 1, podemos observar que las confianzas en favor de las clases difíciles deberían utilizar una REF con un valor de λ pequeño, mientras que aquellas correspondientes a clases más fáciles deberían considerar valores mayores de λ para poder encontrar un buen balance entre sus predicciones.*

En definitiva, tras ver cómo podemos modificar las confianzas dadas por los clasificadores base, de manera similar al WV, agregamos los votos en cada fila:

$$\text{Clase} = \arg \max_{i=1,\ldots,m} \sum_{1 \le j \ne i \le m} \text{REF}_{ij}(r_{ij}, 1) = \arg \max_{i=1,\ldots,m} \sum_{1 \le j \ne i \le m} (r_{ij})^{\lambda_{ij}} \tag{3}$$

donde λ_{ij} es el parámetro que corresponde a $\text{REF}_{ij}(r_{ij}, 1)$. Hacer notar que el caso del WV se recupera cuando $\lambda_{ij} = 1$ para todo $i, j = 1, \ldots, m$ e $i \ne j$.

3.3. Ajustando la agregación para potenciar las clases difíciles

El ajuste de los parámetros es necesario para poder adaptar la agregación a cada clase en cada problema. En este trabajo consideramos el uso del algoritmo

evolutivo de codificación real CHC [7] para optimizar la función objetivo propuesta. A continuación presentamos los dos factores clave de nuestra propuesta como son la propia función objetivo y la codificación de los parámetros.

Función objetivo. Nuestro objetivo debe ser diferente al de la precisión global, ya que esta no tiene en cuenta las clases difíciles. Por ello, proponemos la siguiente función para estimar la calidad de los parámetros $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_{m(m-1)})$ (tenemos un parámetro por cada posición de la matriz de votos):

$$\text{Fitness}(\boldsymbol{\lambda}) = \text{Margen}(\boldsymbol{\lambda}) + 0{,}5 \cdot \text{GM}(\boldsymbol{\lambda}) + 0{,}5 \cdot \text{AvgAcc}(\boldsymbol{\lambda}), \tag{4}$$

donde GM es la media geométrica [4] de las precisiones por clase, AvgAcc es la media aritmética [8] de las precisiones por clase y Margen cuantifica la calidad de la separación de la clases y se calcula como sigue.

$$\text{Margen} = \arg \min_{c=1,\ldots,n_c} \frac{V_i^c - V_j^c}{n_{Tr} \cdot m} \tag{5}$$

donde C_i es la clase predicha y C_j es la segunda clase con más confianza (Eq. (3)); V_i^c, V_j^c son los valores obtenidos en Eq. (3) para cada clase (por la instancia c), respectivamente. n_c es el número de instancias clasificadas correctamente, ya que solo estas son utilizadas. El margen es normalizado para reducir su influencia en la función objetivo. Tomamos el menor de los márgenes ya que es el que mejor representa la separación entre las clases más conflictivas.

Representación de parámetros. Para codificar el conjunto de parámetros reales ($\boldsymbol{\lambda}$) debemos de trasladar el rango de λ que va de 0 a ∞ a unidades codificables en el cromosoma. Utilizamos la siguiente codificación que nos permite buscar en todo el espacio homogéneamente (Figura 1):

$$\Phi(\boldsymbol{\lambda}) = (\phi(\lambda_1), \phi(\lambda_2), \ldots, \phi(\lambda_{m(m-1)})) = (c_{\lambda_1}, c_{\lambda_2}, \ldots, c_{\lambda_{m(m-1)}}) \tag{6}$$

donde cada gen $c_{\lambda_i} \in (0, 1)$, $i \in \{1, \ldots, m(m-1)\}$ y el valor del parámetro se recupera como sigue

$$\lambda_i = \phi^{-1}(c_{\lambda_i}) = \begin{cases} (2 \cdot c_{\lambda_i})^2 & \text{si } c_{\lambda_i} \leq 0{,}5 \\ \dfrac{1}{(2 \cdot (c_{\lambda_i} - 0{,}5))^2} & \text{en otro caso.} \end{cases} \tag{7}$$

4. Marco Experimental

4.1. Clasificadores base y parámetros

Utilizamos SVMs [15] como clasificador base para estudiar la validez de la propuesta. Las confianzas las obtenemos a partir de la estimación de probabilidades del modelo logístico SVM [14]. La Tabla 2 describe los parámetros utilizados en los experimentos que hemos llevado a cabo con la herramienta KEEL [2]. Utilizamos dos configuraciones diferentes con diferentes valores de C y de kernel para estudiar la robustez de la propuesta. Como agregación base para llevar a cabo la comparativa consideramos el método de estimación de probabilidades de Wu *et al.* [17] (PE) cuya precisión ha sido probada [9].

Tabla 2. Especificación de parámetros.

Algoritmo	*Parámetros*
SVM_{Poly}	C = 1.0, Tolerancia = 0.001, Epsilon = 1.0E-12 Kernel = Polinomial, Grado del polinomio = 1
SVM_{Puk}	C = 100.0, Tolerancia = 0.001, Epsilon = 1.0E-12 Kernel = Puk, PukKernel ω = 1.0, PukKernel σ = 1.0
CHC	Tamaño población = 50, Evaluaciones = $1000 \cdot m^2$ Bist por gen (codificación Gray) = 30 Reinicios sin mejora = 3

4.2. Conjuntos de datos y evaluación

Consideramos veintiocho conjuntos de datos de los repositorios UCI [3] y KEEL [1], mostrados en la tabla Tabla 3. Para llevar a cabo la evaluación de los resultados utilizamos la precisión global, la GM y la AvgAcc. En el caso de las clases difíciles, la GM es la que mejor va a reflejar este problema y por tanto la más válida a la hora de evaluarlo. Utilizamos un modelo de validación cruzada de 5 particiones para obtener los resultados. Para llevar a cabo una evaluación

Tabla 3. Descripción de los conjuntos de datos.

C. datos	#Ej.	#Atr.	#Num.	#Nom.	#Cl.	C. datos	#Ej.	#Atr.	#Num.	#Nom.	#Cl.
Balance	625	4	4	0	3	Page-blocks	548	10	10	0	5
Contraceptive	1473	9	9	0	3	Shuttle	2175	9	9	0	5
Hayes-roth	132	4	4	0	3	Autos	159	25	15	10	6
Iris	150	4	4	0	3	Dermatology	358	34	1	33	6
NewThyroid	215	5	5	0	3	Flare	1066	11	0	11	6
Splice	319	60	0	60	3	Glass	214	9	9	0	7
Tae	151	5	5	0	3	Satimage	643	36	36	0	7
Thyroid	720	21	21	0	3	Segment	2310	19	19	0	7
Wine	178	13	13	0	3	Zoo	101	16	0	16	7
Car	1728	6	0	6	4	Ecoli	336	7	7	0	8
Lymphography	148	18	3	15	4	Led7digit	500	7	0	7	10
Vehicle	846	18	18	0	4	Penbased	1100	16	16	0	10
Cleveland	297	13	13	0	5	Yeast	1484	8	8	0	10
Nursery	1296	8	0	8	5	Vowel	990	13	13	0	11

honesta de los resultados, utilizamos test estadísticos no paramétricos tal y como se recomienda en [6,10]. En este caso, dado que realizamos comparativas por pares, utilizaremos el test de Wilcoxon [16].

5. Estudio experimental

En esta sección mostramos la validez de nuestra propuesta basada en REFs (denotada como RA) para potenciar las clases difíciles. Para ello, consideramos dos configuraciones diferentes de las SVM. Los resultados obtenidos con SVM_{Poly} como clasificador base pueden verse en la parte izquierda de la Tabla 4, mientras que los correspondientes a SVM_{Puk} pueden verse en la parte derecha. Los resultados de precisión y AvgAcc aparecen como porcentajes tal y como es habitual. Podemos observar en las tablas como la precisión global se mantiene, mientras que la GM y la AvgAcc han sido aumentadas mediante RA, siendo la mejora en GM notable. En cualquier caso, para obtener conclusiones significativas estos hechos deben ser contrastados con el estudio estadístico adecuado mediante el test de Wilcoxon, cuyos resultados se muestran en la Tabla 5.

Atendiendo a los resultados de los test, las conclusiones obtenidas son similares. Ambos métodos (PE y RA) obtienen precisiones globales equivalentes, pero

Tabla 4. Resultados de precisión, GM y AvgAcc para OVO con PE y RA usando las dos configuraciones de SVM.

	SVM_{Poly}						SVM_{Puk}					
	Accuracy		GM		AvgAcc		Accuracy		GM		AvgAcc	
C. datos	PE	RA	PE	RA	PE	RA	PE	RA	PE	RA	PE	RA
Autos	**74.80**	74,17	**.5479**	,5463	**72.69**	71,73	**68.53**	64,76	**.2544**	,2279	**65.06**	60,19
Balance	90,40	**91.52**	,8310	**.9072**	85,35	**91.12**	88,00	**88.96**	**.8660**	,8514	**86.93**	85,89
Car	92,71	**93.58**	,8651	**.9390**	87,18	**93.97**	63,60	**67.01**	,7452	**.7606**	77,58	**78.88**
Cleveland	**58.25**	46,78	,0000	**.0776**	**30.88**	29,94	45,09	**45.76**	**.0000**	**.0000**	29,78	**30.27**
Contraceptive	49,83	**51.93**	,4604	**.5280**	47,34	**53.19**	**48.41**	45,82	,4406	**.4667**	45,70	**47.29**
Dermatology	**94.13**	**94.13**	**.9408**	**.9408**	**94.58**	**94.58**	**96.09**	94,14	**.9574**	,9238	**96.03**	93,15
Ecoli	**77.69**	74,71	**.1544**	,1418	**68.18**	66,52	**75.31**	**75.31**	,1381	**.1550**	67,35	**68.34**
Flare	**74.67**	73,45	,4517	**.6008**	61,02	**66.10**	**69.42**	65,39	,3277	**.5177**	59,43	**60.42**
Glass	61,26	**61.68**	,2045	**.4834**	55,40	**65.07**	70,60	**72.95**	,5372	**.5598**	68,04	**69.81**
Hayes-Roth	52,22	**71.94**	,4985	**.7015**	55,05	**72.46**	79,54	**81.82**	,8072	**.8277**	82,30	**84.24**
Iris	**96.00**	**96.00**	,9580	**.9583**	**96.00**	**96.00**	94,00	**94.67**	,9375	**.9442**	94,00	**94.67**
Led7digit	**73.00**	**73.00**	,7110	**.7145**	73,01	**73.07**	70,20	**71.00**	,6840	**.6959**	70,32	**71.22**
Lymphography	81,68	**83.06**	,3348	**.3449**	64,87	**66.54**	**80.34**	**80.34**	**.1557**	**.1557**	**54.98**	**54.98**
NewThyroid	**97.21**	96,74	,9599	**.9667**	96,16	**96.83**	**97.67**	**97.67**	**.9811**	**.9811**	**98.16**	**98.16**
Nursery	**91.90**	91,74	,6529	**.7237**	82,22	**87.84**	**81.33**	80,86	**.6793**	,6679	**82.28**	81,82
Pageblocks	**94.70**	87,40	,3042	**.6602**	68,23	**80.39**	94,16	**94.34**	,2757	**.2837**	67,40	**68.66**
Penbased	95,27	**95.36**	,9513	**.9525**	95,29	**95.40**	**97.82**	**97.82**	**.9781**	**.9781**	**97.85**	**97.85**
Satimage	**84.14**	82,89	,7703	**.7874**	79,55	**80.14**	84,92	**85.39**	,8315	**.8368**	84,16	**84.69**
Segment	92,55	**94.63**	,9197	**.9444**	92,55	**94.63**	97,10	**97.14**	,9704	**.9708**	97,10	**97.14**
Shuttle	**96.37**	95,36	**.3477**	,3439	80,67	**80.87**	**99.72**	98,62	**.7650**	,7615	**93.14**	92,76
Splice	79,59	**80.22**	,8325	**.8374**	84,29	**84.69**	**64.56**	58,02	,3787	**.5586**	51,44	**66.42**
Tae	51,72	**54.41**	,4869	**.5220**	51,91	**54.24**	56,30	**58.28**	,5513	**.5707**	56,24	**58.18**
Thyroid	95,69	**97.22**	,4445	**.8816**	67,88	**89.39**	**92.64**	92,50	,4971	**.5559**	62,44	**68.47**
Vehicle	72,46	**73.40**	**.6970**	,6956	72,82	**73.84**	80,49	**80.61**	,7873	**.7887**	80,71	**80.83**
Vowel	69,90	**71.11**	,6822	**.6921**	69,90	**71.11**	**99.39**	**99.39**	**.9936**	**.9936**	**99.39**	**99.39**
Wine	**97.16**	**97.16**	**.9684**	**.9684**	**96.99**	**96.99**	**98.30**	**98.30**	**.9857**	**.9857**	**98.60**	**98.60**
Yeast	**59.10**	54,65	**.0000**	**.0000**	**56.74**	55,46	**56.54**	55,12	,0000	**.0905**	55,37	**55.73**
Zoo	95,05	**97.00**	,0000	**.4000**	85,24	**91.43**	84,19	**90.10**	**.0000**	**.0000**	64,05	**72.14**
Mean	80,34	**80.54**	,5706	**.6521**	74,00	**77.63**	**79.80**	79,72	,5902	**.6111**	74,49	**75.72**

Tabla 5. Wilcoxon test para comparar RA y PE. R^+ corresponde a la suma de rangos para RA y R^- para PE.

	SVM_{Poly}				SVM_{Puk}			
Medida	R^+	R^-	Hipótesis ($\alpha = 0{,}05$)	p-valor	R^+	R^-	Hipótesis ($\alpha = 0{,}05$)	p-value
Precisión	230,0	176,0	No rechazada	0,511089	219,5	186,5	No rechazada	0,883838
GM	367,0	39,0	Rechazada para RA	0,000194	305,0	101,0	Rechazada para RA	0,027306
AvgAcc	353,0	53,0	Rechazada para RA	0,000808	313,5	92,5	Rechazada para RA	0,017668

RA obtiene resultados significativamente mejores en términos de GM y AvgAcc con p-valores muy bajos para ambas configuraciones de SVM. Por tanto, además de mejorar las precisiones sobre las clases difíciles podemos decir que estamos ante un método robusto ante las diferentes configuraciones utilizadas. Únicamente destacar que la mejora obtenida es mayor en SVM_{Poly} debido a que las confianzas dadas por esta configuración son mejores (están mejor repartidas en el rango por la configuración del parámetro C) y por tanto dan más información sobre el proceso de clasificación.

6. Conclusiones

En este trabajo hemos puesto de manifiesto el problema de las clases difíciles en la estrategia OVO. Para tratar de mejorar la clasificación sobre estas clases hemos propuesto una nueva metodología de agregación que generaliza al voto ponderado.

Esta metodología permite aprender los parámetros de las REFs para diferenciar mejor las clases difíciles, lo que ha quedado empíricamente probado con las diferencias estadísticas encontradas en términos de GM.

Agradecimientos Este trabajo ha sido parcialmente subvencionado por el Ministerio de Educación y Ciencia bajo los proyectos TIN2010-15055 y TIN2011-28488 y el plan Andaluz de investigación P10-TIC-6858.

Referencias

1. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. J. Mult.-Valued Logic Soft Comput. 17, 255 – 287 (2011)
2. Alcalá-Fdez, J., Sánchez, L., García, S., del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., Fernández, J., Herrera, F.: KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput. 13(3), 307–318 (2009)
3. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
4. Barandela, R., Sánchez, J.S., García, V., Rangel, E.: Strategies for learning in class imbalance problems. Pattern Recogn. 36(3), 849–851 (2003)
5. Bustince, H., Barrenechea, E., Pagola, M.: Restricted equivalence functions. Fuzzy Sets Syst. 157(17), 2333–2346 (2006)
6. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7, 1–30 (2006)
7. Eshelman, L.J., Schaffer, J.D.: Real-coded genetic algorithms and interval-schemata. In: Whitley, D.L. (ed.) Foundation of Genetic Algorithms 2. pp. 187–202. Morgan Kaufmann, San Mateo, CA (1993)
8. Ferri, C., Hernández-Orallo, J., Modroiu, R.: An experimental comparison of performance measures for classification. Pattern Recogn. Lett. 30(1), 27–38 (2009)
9. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. Pattern Recogn. 44(8), 1761 – 1776 (2011)
10. García, S., Herrera, F.: An extension on "statistical comparisons of classifiers over multiple data sets"for all pairwise comparisons. J. Mach. Learn. Res. 9, 2677–2694 (2008)
11. Hüllermeier, E., Vanderlooy, S.: Combining predictions in pairwise classification: An optimal adaptive voting strategy and its relation to weighted voting. Pattern Recogn. 43(1), 128–142 (2010)
12. Kuncheva, L., Whitaker, C., Shipp, C., Duin, R.: Limits on the majority vote accuracy in classifier fusion. Pattern Anal. App. 6, 22–31 (2003)
13. Lorena, A.C., Carvalho, A.C., Gama, J.M.: A review on the combination of binary classifiers in multiclass problems. Artif. Intell. Rev. 30(1-4), 19–37 (2008)
14. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in Large Margin Classifiers. pp. 61–74. MIT Press (1999)
15. Vapnik, V.: Statistical Learning Theory. New York: Wiley (1998)
16. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bull. 1(6), 80–83 (1945)
17. Wu, T.F., Lin, C.J., Weng, R.C.: Probability estimates for multi-class classification by pairwise coupling. J. Mach. Learn. Res. 5, 975–1005 (2004)